

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 11-282815

(43)Date of publication of application : 15.10.1999

(51)Int.Cl. G06F 15/16
 G06F 15/16
 G06F 9/46
 G06F 12/08

(21)Application number : 10-103970 (71)Applicant : NEC CORP

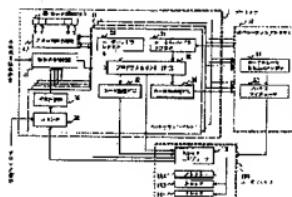
(22)Date of filing : 31.03.1998 (72)Inventor : SAKAI JUNJI

(54) MULTI-THREAD COMPUTER SYSTEM AND MULTI-THREAD EXECUTION CONTROL METHOD

(57)Abstract:

PROBLEM TO BE SOLVED: To provide a multi-thread computer system which includes a user level interrupt mechanism of reduced overhead and a fast exclusive control mechanism to execute a fine grain multi-thread.

SOLUTION: When a thread execution is started by a processor element 11, the value of a counter 26 is decreased for every clock. When the value of the counter 26 is set to zero, an interrupt control part 41 starts the user level interrupt processing to shift its control to the address of a user thread scheduler 101 that is set at a user handler register 20. Meanwhile, a test-and-set instruction performs a lock operation to a lock variable set 30 included in a processor 1 and attains an exclusive control among the elements 11.



LEGAL STATUS

[Date of request for examination] 31.03.1998

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number] 3546694

[Date of registration] 23.04.2004

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's

(19) 日本国特許庁 (JP)

(12) 公開特許公報 (A)

（11）特許出願公願番号

特開平11-282815

(43) 公開日 平成11年(1999)10月15日

(51) Int.Cl.
G 0 6 F 15/16

識別語

四

G 0 6 F 15/16
9/46
12/08

430B
390Z
360B
H

Q

審査請求 有 請求項の数9 ED (全 9 頁)

(21)出願番号 筆頭平10-103970

(22) 出願日 平成10年(1998) 3月31日

© 2013 IEEE. All rights reserved.

日本體育出版社

日本電気株式会社
東京都港区若木二丁目名番1号

(22) 晴明者 酒井 流麗

四升 浮動

東京都港区芝五丁目7番1号 日本電気株式会社

式会社内

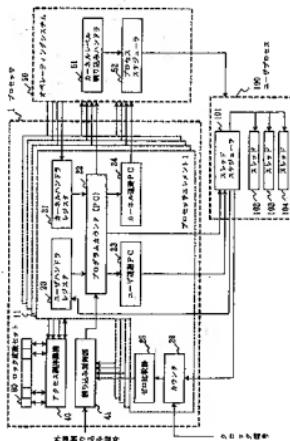
(74) 代理人 介理士 境 廣已

(54)【発明の名前】マルチスレッド計算機システム及びマルチスレッド計算機システム

(ETM 重绘)

【課題】 粒度マルチスレッド実行を可能にするための、オーバーヘッドの少ないユーザレベル割り込み機構と高速な排他制御機構とを備えたマルチスレッド計算機システムを提供する。

【解決手段】 プロセッサエレメント 1 1 でスレッドの実行が始まるとカウンタ 2 6 はクロック毎に減じられる。カウンタ 2 6 の値がゼロになると書け込み制御部 4 がユーザーベースの書け込み処理を開始し、ユーザハンドルレジスタ 2 0 に設定されているユーザースレッドスケジューラ 1 0 1 のアドレスに制御を移す。また、テストアンドセット命令はプロセッサ 1 内のロック変数セット 3 0 に対してロック操作を行い、プロセッサエレメント間での操作権を変更する。



【特許請求の範囲】

【請求項1】 ユーザレベルの割り込みハンドラのエントリアドレスを保持するユーザハンドラレジスタと、プログラムカウンタの値の遮還先レジスタであるユーザ遮還P Cと、

割り込み要求信号の入力によりプログラムカウンタ値をユーザ遮還P Cに設定すると共にユーザハンドラレジスタ値を新しいプログラムカウンタに設定するような割り込み処理を行なう割り込み制御部とを備え、

複数のスレッドを管理する際のプリエンプションを上記割り込み機構を用いてユーザ空間内で行なうことを特徴とするマルチスレッド計算機システム。

【請求項2】 ロック毎に値が減じられていくダウンカウンタと、カウンタ値がゼロになった場合に信号を山出するゼロ比較器と、ゼロ比較器のゼロ一致(山出)によってユーザレベル割り込み処理を行なう割り込み制御部を備える請求項1記載のマルチスレッド計算機システム。

【請求項3】 各プロセッサエレメントからアクセスできる1ビットの記憶装置の集合と、その各記憶装置に対してプロセッサエレメント間で排他的に値の操作を行なう計算機命令セットとを備える請求項1または2記載のマルチスレッド計算機システム。

【請求項4】 各プロセッサエレメントに対応した識別番号をもつて1個を到着順に格納するキュー構造の集合と、キュー構造に対してプロセッサエレメント間で排他的にトーケンの追加、検索あるいは削除の操作を行なうキュー制御機構と、前記キュー構造に対するトーケンの追加、検索あるいは削除操作を行なうための計算機命令セットとを備えることを特徴とする請求項1または2記載のマルチスレッド計算機システム。

【請求項5】 各プロセッサエレメントで共有されるキャッシュメモリ、

キャッシュメモリ中の同一アドレスに対するアクセスをプロセッサエレメント間で調停するアクセス調停機構と、

キャッシュメモリ上にあるメモリ要素に対する排他的な値の操作を行なう計算機命令セットとを備えることを特徴とする請求項1または2記載のマルチスレッド計算機システム。

【請求項6】 複数のプロセッサエレメントを含むプロセッサと前記複数のプロセッサエレメントで共有される主記憶メモリを備え、一つのユーザプロセスを複数のスレッドに分割し、そのユーザプロセス内のスレッドスケジューラの制御の下に複数のスレッドを複数のプロセッサエレメントに割り当てる同時に実行するマルチスレッド計算機システムにおけるマルチスレッド実行制御方

法において、

(a) ユーザプロセスのスレッドが割り当てるプロセッサエレメント内のユーザハンドラレジスタにそのユーザプロセスのスレッドスケジューラの置かれているメモリアドレスを設定すると共に、そのプロセッサエレメント内のカウンタにそのスレッドに割り当てるタイムクォンタム値を設定する段階

(b) プロセッサエレメントにおけるスレッドの実行開始と同時にそのプロセッサエレメント内の前記カウンタの値を一定周期で更新し、予め定められたカウント値に達した時点でユーザレベル割り込みを発生させる段階

(c) ユーザレベル割り込みの処理において、割り込み要求元のプロセッサエレメントの現在のプログラムカウンタの値をそのプロセッサエレメント内のユーザ遮還P Cに設定し、そのプロセッサエレメント内のユーザハンドラレジスタに設定されたメモリアドレスをプログラムカウンタに設定することにより制御をユーザプロセス内のスレッドスケジューラに移す段階を含むことを特徴とするマルチスレッド実行制御方法。

【請求項7】 複数のプロセッサエレメントで実行される複数のスレッド間の排他制御を、プロセッサ内に設けられ且つプロセッサエレメント間で排他的に値の操作を行なえる計算機命令セットによって各プロセッサエレメントからアクセス可能な1ビットの記憶装置の集合を用いて行なうことを特徴とする請求項6記載のマルチスレッド実行制御方法。

【請求項8】 複数のプロセッサエレメントで実行される複数のスレッド間の排他制御を、プロセッサ内に設けられたプロセッサ内のキュー構造の集合であって、プロセッサエレメント間で排他的にトーケンの追加、検索あるいは削除を行なえる計算機命令セットによって各プロセッサエレメントからアクセスできるキュー構造の集合を用いて行なうことを特徴とする請求項6記載のマルチスレッド実行制御方法。

【請求項9】 複数のプロセッサエレメントで実行される複数のスレッド間の排他制御を、同一アドレスに対するアクセスをプロセッサエレメント間で調停するアクセス調停機構を通じて各プロセッサエレメントからアクセス可能な共有キャッシュメモリ上のメモリ要素であって、プロセッサエレメント間で排他的に値の操作が行なえる計算機命令セットによってアクセス可能なメモリ要素を用いて行なうことを特徴とする請求項6記載のマルチスレッド実行制御方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は並列計算機システムに関する、特に、共有メモリ型マルチプロセッサ計算機上で、オペレーティングシステムを介すことなく複数スレッドを効率的にスケジューリングするマルチスレッド

計算機システムに関する。 3

【 0 0 0 2 】

【 従来の技術】より高い演算処理性能を得るために、一つのシステム内に複数のプロセッサエレメントを備えるマルチプロセッサ構成の計算機システムがある。そのような計算機システムのうち、各プロセッサエレメントが主記憶メモリを共有する構成のものは共有メモリ型マルチプロセッサ計算機システムと呼ばれ、分散メモリ型システムに比べプログラムの記述が容易であるという利点を持つ。

【 0 0 0 3 】他方、ソフトウェア面では、一つのプロセスをスレッドと呼び、制御の流れに分割し、複数のスレッドを並行して実行する、マルチスレッド実行と呼ばれる並列実行方式がある。マルチプロセッサ計算機システム上では、複数のスレッドを複数のプロセッサエレメントに割り当てる同時に実行させることで処理性能が向上する。

【 0 0 0 4 】マルチプロセッサ計算機システムにおけるスレッドの管理は、通常、マルチプロセッサ用オペレーティングシステムが行なう。すなわち、新しいスレッドの生成、別のスレッドとの同期、スレッドの消滅等を行なう場合、当該スレッドはオペレーティングシステムのサービスを呼び出す。

【 0 0 0 5 】スレッドのプリエンプションもオペレーティングシステムを介して行なわれる。プリエンプションとは、長時間プロセッサを占有し続けるスレッドの実行を中断させ、他のスレッドの実行に切り替える処理である。プリエンプション機能をもつ計算機システムは、ハードウェアタイマ装置と、所定の時間が経過するとタイマ装置からプロセッサに割り込み信号を伝える仕組みを持っている。ユーザプログラムの実行を開始してから一定の時間が経過すると、タイマ割り込みによってオペレーティングシステムに制御が移り、オペレーティングシステムが実行スレッドの切り替えを行なう。

【 0 0 0 6 】マルチプロセッサ計算機システム上で複数のスレッドを効率良く実行するには、スレッドの生成、同期、切り替え、消滅といったスレッド管理のオーバーヘッドを低減させることが重要である。

【 0 0 0 7 】サン・マイクロシステムズのソラリス (Solaris) オペレーティングシステムでは、オペレーティングシステムではなく、ユーザレベルのライブラリによってスレッドの管理を行なうことができる。これにより、より低いオーバーヘッドでスレッド管理ができるが、プリエンプションの場合にはオペレーティングシステムが介在する（マルチスレッドプログラミング入門、アスキー出版、1996年9月、p69 ~ p70 および p76 ~ p77）。

【 0 0 0 8 】特開平5-158900号公報に記載されているプリエンプション処理回路は、オペレーティングシステムではなく、専用ハードウェアによってプリエンプションの受け付けを行なうものであるが、受け付け後の切り替

え処理はオペレーティングシステムへの通常の割り込みによって行なっている。

【 0 0 0 9 】他方、複数のスレッドによって共有される計算機資源、例えば主記憶メモリ上の共有変数を読み書きする場合には、複数のスレッドがその計算機資源を同時にアクセスすることで競合した処理結果を得ることを回避するために、排他制御と呼ばれる処理を行なう必要がある（排他制御については、例えば、A. S. タンネンバウム原著「OSの基礎と応用」ブレンティスピール、ト・ト・バシン、第2章、2、2節等に説明されている）。

【 0 0 1 0 】排他制御は通常、テストアンドセットやエクスチェンジといった排他制御用のプロセッサ命令によって実現される。これらの命令は、主記憶上のあるメモリセルの値の検査とそのメモリセルへの新しい値の設定とを不可分けに行なう。例えば、テストアンドセット命令は、以下の3つの操作を、割り込みあるいは他のプロセッサによって分断されること無く行なう。

(1) 主記憶上のあるメモリセルの値の計算機レジスタへの読み出し

(2) そのメモリセルへの値の書き込み

(3) 計算機レジスタへ読み出した値と値0との比較

【 0 0 1 1 】共有メモリ型マルチプロセッサ計算機システムでは、異なるプロセッサ間の排他制御を行なうため、これらの排他制御用命令は共有している主記憶メモリに対して作用する。すなわち、排他制御自身のために主記憶メモリへのアクセスが必要になる。

【 0 0 1 2 】

【 発明が解決しようとする課題】従来技術の第1の問題点は、マルチスレッド実行のために十分高速な切り込み応答処理ができず、スレッド切り替えのオーバーヘッドが大きかった点である。これは、切り込み処理がオペレーティングシステム内で処理されるため、ユーザプログラムからオペレーティングシステムへのコンテキストの切り替えが発生するからである。

【 0 0 1 3 】第2の問題点は、排他制御処理自身のために主記憶メモリへのアクセスが発生し、マルチスレッド処理に不可欠なスレッド間の排他制御や同期処理のオーバーヘッドが大きかった点である。近年のプロセッサの処理速度は主記憶メモリのアクセス速度よりも格段に速いため、多くのプロセッサはキャッシュメモリと呼ばれる高速で小容量のメモリをプロセッサと主記憶メモリの間に備え、プロセッサは専らキャッシュメモリをアクセスすることでメモリアクセスによる処理の遅延を回避している。しかし、排他制御命令は常に主記憶メモリへのアクセスを行なうため、処理速度の低下を招く。

【 0 0 1 4 】上に挙げた問題点はいずれもマルチスレッド処理のオーバーヘッドを増大させる原因となり、特に、粒度の小さいスレッド（含有する命令数の少ないスレッド）を単位としてマルチスレッド処理を行なう場合に大きな影響をもたらす。

【 0 0 1 5 】

【発明の目的】本発明の目的は、オペレーティングシステムが介在しない高速なユーザレベル割り込みを提供し、マルチスレッド処理の効率を向上させることである。

【 0 0 1 6 】本発明の他の目的は、マルチプロセッサ計算機システムにおけるプロセッサエレメント間の高速なロック機構を提供し、マルチスレッド処理の効率を向上させることである。

【 0 0 1 7 】

【課題を解決するための手段】本発明のマルチスレッド計算機システムは、オペレーティングシステムが介在しない高速なユーザレベル割り込みを提供する高速ユーザレベル割り込み機構を有する。具体的には、プロセッサ内の割り込み制御前(図2の4.1)と、各プロセッサエレメント内のカウント(図2の2.6)と、ユーザハンドレジスタ(図2の2.0)とを備える。

【 0 0 1 8 】また、本発明のマルチスレッド計算機システムは、マルチプロセッサ計算機システムにおけるプロセッサエレメント間の高速なロック機構を提供する高速ロック機構を有する。具体的には、第1の高速ロック機構にあっては、プロセッサ内にプロセッサエレメント間で共有されるロック変数セット(図2の3.0)と、それを操作するための計算機命令を備える。また、第2の高速ロック機構にあっては、キュー構造のロック変数(図3の6.1、6.2及び6.9)と有する。さらに第3の高速ロック機構にあっては、プロセッサ内の共有キャッシュメモリ(図4の7.1)と、キャッシュ上のメモリ領域を操作するための計算機命令とを有する。

【 0 0 1 9 】また本発明のマルチスレッド実行制御方法は、複数のプロセッサエレメントを含むプロセッサと前記複数のプロセッサエレメントで共有される主記憶メモリとを備え、一つのユーザプロセスを複数のスレッドに分割し、そのユーザプロセス内のスレッドスケジューラの制御の下に複数のスレッドを複数のプロセッサエレメントに割り当てる同時に実行するマルチスレッド計算機システムにおけるマルチスレッド実行制御方法において、オペレーティングシステムが介在しない高速なユーザレベル割り込みを提供するために、以下の段階を含んでいる。

(a) ユーザプロセスのスレッドが割り当てるプロセッサエレメント内のユーザハンドレジスタにそのユーザプロセスのスレッドスケジューラの置かれているメモリアドレスを設定すると共に、そのプロセッサエレメント内のカウントにそのスレッドに割り当てるタイムクォンタム値を設定する段階

(b) プロセッサエレメントにおけるスレッドの実行開始と共にそのプロセッサエレメント内の前記カウントの値を一定周期で更新し、予め定められたカウント値に達した時点でユーザレベル割り込みを発生させる段階

(c) ユーザレベル割り込みの処理において、割り込み要求元のプロセッサエレメントの現在のプログラムカウンタの値をそのプロセッサエレメント内のユーザ逃避PCに設定し、そのプロセッサエレメント内のユーザハンドレジスタに設定されたメモリアドレスをプログラムカウンタに設定することにより制御をユーザプロセス内のスレッドスケジューラに移す段階

【 0 0 2 0 】また、本発明のマルチスレッド実行制御方法は、マルチプロセッサ計算機システムにおけるプロセッサエレメント間の高速な排他制御を可能にするために、複数のプロセッサエレメントで実行される複数のスレッド間の排他制御を、以下の(a)、(b)、(c)の何れかによって実現している。

(a) プロセッサ内に設けられ且つプロセッサエレメント間で排他的に値の操作が行える計算機命令セットによって各プロセッサエレメントからアクセス可能な1ビットの記憶装置の集合を用いて排他制御する。

(b) プロセッサエレメントに対応した識別番号をもつトーカンを到着順に格納するプロセッサ内のキュー構造の集合であって、プロセッサエレメント間で排他的にトーカンの追加・検索あるいは削除が行なえる計算機命令セットによって各プロセッサエレメントからアクセスできるキュー構造の集合を用いて排他制御する。

(c) 同一アドレスに対するアクセスをプロセッサエレメント間で調停するアクセス制御機構を通じて各プロセッサエレメントからアクセス可能な共有キャッシュメモリ上のメモリ要素であって、プロセッサエレメント間で排他的に値の操作が行なえる計算機命令セットによってアクセス可能なメモリ要素を用いて排他制御する。

【 0 0 2 1 】

【作用】カウントがゼロになると割り込み制御部はユーザレベル割り込み処理を開始する。ユーザレベル割り込み処理では、予めユーザプロセスのスレッドスケジューラの開始アドレスに設定しておいたユーザハンドレジスタの値をプログラムカウンタに設定し、オペレーティングシステムを経由すること無くスレッドスケジューラに制御を移行する。

【 0 0 2 2 】プロセッサ内に備えられた高速ロック機構は、主記憶メモリへアクセスする必要のないロック獲得及び解放機能を提供する。

【 0 0 2 3 】

【発明の実施の形態】

【構成の説明】本発明の実施の形態について、図面を参照して詳細に説明する。

【 0 0 2 4 】図1は、本発明が実施される計算機構成の一例を示す図である。プロセッサ1は内部に複数のプロセッサエレメント1.1、1.2及び1.9を持ち、それらのプロセッサエレメントは共通の主記憶メモリ2に対してアクセスする。

【 0 0 2 5 】図2を参照すると、本発明の第1の実施の

形態は、内部に複数のプロセッサエレメントを有するプロセッサ1と、そのプロセッサ上で動作するオペレーティングシステム50と、そのオペレーティングシステム上で動作するユーザプロセス100とから構成される。【0026】プロセッサ1内の各プロセッサエレメントは、一般的な計算機のプロセッサが持つレジスタ集合や演算ユニット、制御ユニット等の他に、ユーザレベルのブリエンプション割り込みを発生させるためのカウンタ26とゼロ比較器25、ユーザレベル割り込み発生時の制御移動に使用されるユーザハンドレジスタ20、ユーザ退避PC23を備えている。

【0027】カウンタ26の値は、クロック信号によってクロック毎に減算される。ゼロ比較器25は、カウンタ26の値とゼロとを比較する。ユーザハンドレジスタ20はユーザレベル割り込み発生時に実行制御を移すべきプログラムカウンタ値を保持しているレジスタであり、ユーザ退避PC23はユーザレベル割り込みの発生直前のプログラムカウンタ値を退避しておくためのレジスタである。

【0028】図2のプログラムカウンタ(P C)22及び割り込み制御部41は一般的な計算機のプロセッサが持つプログラムカウンタ及び割り込み制御部と同等の機能を有するものであり、カーネルハンドレジスタ21及びカーネル退避PC24は一般的な計算機のプロセッサにおける通常の割り込み発生時に用いられる割り込みハンドレジスタ及びプログラムカウンタ退避用レジスタと同等の機能を有するものである。

【0029】ユーザプロセス100はオペレーティングシステム50により生成される実行プログラムの実体である。ユーザプロセス100は主記憶メモリ空間を共有する複数のスレッド102、103及び104から構成され、これらのスレッドがプロセッサエレメントに割り当てられ実行される。スレッドスケジューラ101はユーザプログラムとリンクされた形でユーザプロセス100の内部に存在し、当該ユーザプロセスを構成する全スレッドの管理を行なう。

【0030】また、プロセッサ内には全プロセッサエレメントで共有されるロック変数セット30とアクセス調停機構40がある。ロック変数セット30は1ビットの状態を記憶できるロック変数の集合であり、アクセス調停機構40はロック変数へのアクセスをプロセッサエレメント間で調停する機構である。

【0031】

【動作の説明】図2を参照して、本実施の形態の動作について詳細に説明する。

【0032】オペレーティングシステム50がユーザプロセス100を開始させると、ユーザプロセス100はその内部に組み込んだスレッドスケジューラ101を呼び出す。スレッドスケジューラ101は、ある決まったスケジューリングアルゴリズムに従って当該ユーザプロ

セス100内のスレッドの集合をスケジューリングする。スレッドスケジューラ101が次に実行すべきスレッド(ここではスレッド102とする)を選択すると、スレッドスケジューラ101はスレッド102に割り当てるタイムクォンタム値をスレッド102が割り当てるプロセッサエレメント11のカウンタ26に設定し、スレッドスケジューラ101の置かれているメモリアドレスを当該プロセッサエレメント11のユーザハンドレジスタ20に設定した後、当該プロセッサエレメント11でスレッド102の実行を開始させる。

【0033】スレッド102が実行されるの並行し

て、カウンタ26の値がクロック信号によって一定時間毎に減らされてゆく。その値がゼロになると、ゼロ比較器25が割り込み制御部41に割り込み要求信号を送る。割り込み制御部41は当該プロセッサエレメント11にてユーザレベル割り込みの処理を開始する。

【0034】ユーザレベル割り込み処理が開始すると、プロセッサエレメント11は現在のプログラムカウンタ22の値をユーザ退避PC23に設定し、ユーザハンドレジスタ20の値をプログラムカウンタ22に設定する。これにより、オペレーティングシステム50を介すことなく、ユーザプロセス100の内部にあるスレッドスケジューラ101に速やかに制御が移行する。スレッドスケジューラ101は、レジスタセットの値をはじめとするスレッド102の実行状態をスレッドスケジューラ101の内部の管理データ領域に保存し、定められたスケジューリングアルゴリズムに従って次に実行開始すべきスレッドを選択し、次スレッドのレジスタセットの値を復元すると共に、カウンタ26にそのスレッドに対応するタイムクォンタム値を設定して次スレッドの実行を開始する。

【0035】これに対して、従来の割り込み処理を用いる場合を以下に説明する。まず、割り込み処理開始時にユーザプログラムからオペレーティングシステムにコンテキストを切り替える。このコンテキスト切り替えには、

○現在のプログラムカウンタ22の値のカーネル退避PC24への設定

○カーネルハンドレジスタ21の値のプログラムカウンタ22への設定

○プロセッサの動作モードのカーネルモードへの切り替え

○必要なレジスタセットの内容の主記憶メモリへの退避といった処理が含まれる。このコンテキスト切り替えによって制御はオペレーティングシステム50内部のカーネルレベル割り込みハンドラ51、次いでプロセススケジューラ52に移行する。プロセススケジューラ52は割り込み要因を分析し、あらかじめ定められたスケジューリングアルゴリズムに従って次に実行すべきプロセスを決定する。そして、再度コンテキスト切り替えを行な

つて次プロセスの実行を開始する。このコンテキスト切り替えには、

○次プロセスのためのレジスタセットの設定

○次プロセス用論理アドレス空間の設定

○プロセッサの動作モードのユーザモードへの切り替え処理

が含まれる。これら一連の処理を終えて、ようやく制御がユーザプロセスに移行する。

【0036】つまり、ユーザレベル割り込み機構の導入により、オペレーティングシステムを経由することによるオーバーヘッドの無い、従来より高速なブリエンブション機構をユーザプログラムで利用することが可能になる。

【0037】なお、ユーザレベル割り込み機構の割り込み要因は、ダウンカウンタのゼロ一致によるブリエンブション割り込みに限定されるものではない。割り込み制御部さえ対応させれば、例えば、ユーザプロセス内で無効命令割り込みを処理するなどの目的で利用することも可能である。

【0038】次に図2を参照してロック変数セット30を用いた高速ロック機構について説明する。各プロセッサエレメントは、本プロセッサに備わっているテストアンドセット命令を用いてロック変数セット30の内部にあるロック変数にアクセスする。あるプロセッサエレメント11がロック変数に対してテストアンドセット命令を用いると、アクセス制御機構40は他のプロセッサエレメントからロック変数へのアクセスを一時的に禁止し、その間にプロセッサエレメント11はロック変数の値をゼロと比較してプロセッサエレメント11内の状態フラグに反映させ、ロック変数に値1を設定する。ソフトウェア側から見ると、本実例で述べたテストアンドセット命令の動作は、従来のプロセッサが備えているテストアンドセット命令の動作と同様であるが、大きな遅延をもたらす主記憶モリへのアクセスを伴わないため、従来よりもはるかに少ないオーバーヘッドで変数をロックすることができる。

【0039】プロセッサ内に搭載するロック変数の個数は8個、16個または32個が適当である。その理由は、ユーザ向けの高機能な排他制御及び同期制御機構はより基本的な排他制御機構を用いて容易に構築可能であり、そのような基本的な排他制御機構が用いるロック変数は少数で済むためと、ロック変数の個数を小さな2の乗乗数とすることでロック変数の番号を指定するための命令フィールド幅を小さくし、命令効率を高めることができるものである。また、ロック変数の個数をレジスタセットの一般的なビット数である32あるいは4以内にすることにより、ロック変数セット全体を一つの特殊レジスタとして扱うことが可能になる。これによりロック変数セットをプロセスに対応するコンテキストに含めることが可能になり、プロセス毎に論理的に独立し

た高速ロック機構を用いることができるようになる。

【0040】上に述べたユーザレベル割り込み機構及び高速ロック機構の導入により、従来より細かな単位での並列処理が可能になり、マルチプロセッサ計算機の実効性能を高めると共に、その適用分野を拡大することができる。これが本実施の形態の効果である。

【発明の他の実施の形態】次に、ユーザレベル割り込み機構の他の実施形態について説明する。

【0042】図2において、カウンタ26の入力はクロック信号となっているが、この信号は計算機システムのクロック信号そのものには限らない。カウンタ26の入力をとしてシステムクロック信号を適当な分周器で数分の一に分周したものを用いれば、カウンタ26を構成するハードウェアのビット数やカウンタ26に値を設定する計算機命令[1]のカウンタ値フィールドのビット幅を減らすことができ、ハードウェア及びソフトウェア双方の効率を改善できる。また、システムクロック信号ではなく、当該プロセッサエレメントの命令実行制御機構の信号をカウンタ26に入力し、1命令実行する毎にカウンタ26が減じられるようにする方式としてもよい。

【0043】次に、高速ロック機構の他の実施形態について説明する。

【0044】図3を参照すると、高速ロック機構の第2の実施の形態は、各ロック変数がキュー構造になっているものである。各キュー6-1、6-2及び6-9には、プロセッサエレメント数に等しい個数までのトークンを格納することができる。トークンは各プロセッサエレメントが発行するもので、トークンには発行したプロセッサエレメントの番号が付される。キューに一つ以上のトークンがある場合、当該キューの先頭にあるトークンを発行したプロセッサエレメントがそのキューに応じるロック変数をロックしていることを意味する。このロック変数を操作するために、ロック試行命令とロック解除命令の2種の計算機命令が備わっている。プロセッサエレメント11、12、…、19とキュー6-1、6-2、…、6-9との間に設けられたキュー制御機構60は、ロック試行命令、ロック解除命令の実行時、キュー6-1、6-2、…、6-9に対してプロセッサエレメント間で排他的にトークンの追加、検索、削除の操作を行なう。

【0045】プロセッサエレメント11がロック試行命令を実行すると、キュー制御機構60は、ロック試行命令のオペランドで指定されたキュー内(ここではキュー6-1とする)に当該プロセッサエレメント11のトークンが存在するか否かを検査し、もしトークンが存在しなければ新たにトークンを投入する。また、キュー制御機構60は指定されたキュー6-1の先頭に当該プロセッサエレメント11に対応するトークンがあるか否かを調べ、その結果をプロセッサエレメント11の状態フラグに設定する。ユーザプログラムはロック試行命令に続

て状態フラグを参照する分岐命令を実行することで、ロックを獲得できたか否かを判断する。

【 0 0 4 6 】 プロセッサエレメント 1 1 がロック解除命令を実行すると、キュー制御機構 6 0 は、ロック試行命令のオーバーランドで指定されたキュー内(ここでキュー 6 1 とする)に当該プロセッサエレメント 1 1 のトークンがあるか否かを検査し、もしあればそのトークンをキュー 6 1 から除去する。

【 0 0 4 7 】 本実施形態は、ロックを要求した順序をキュー内で記憶し先着順にロックを獲得させるという点で、より公平な併用制御機構を実現できる特長を持つ。

【 0 0 4 8 】 図4を参照すると、高速ロック機構の第3の実施の形態は、ロック変数のための特別なレジスタ等を持たず、プロセッサ内のキャッシュメモリ(共有キャッシュ) 7 1 上のメモリ領域をロック変数として利用するものである。キャッシュメモリ 7 1 と各プロセッサエレメント 1 1 、1 2 及び 1 9 の間にはアクセス調停機構 7 0 があり、同一メモリアドレスに対するアクセスを調停する。プロセッサは、従来からの主記憶メモリに対する

ロック処理命令に加え、キャッシュ 7 1 上のメモリ領域をロックするための特別なテストアンドセット命令を備える。従来からの主記憶メモリに対するロック処理命令の場合、ロック変数の値を設定する際にキャッシュラインの主記憶メモリへの吐き出しを行なうが、この特別なテストアンドセット命令は通常のメモリアクセス命令と同様に処理される。即ち、ロック変数値を設定する際にキャッシュラインの主記憶メモリへの吐き出しを行なわない。このため、通常のキャッシュ操作によってロック変数がキャッシュから主記憶メモリへ戻り出されない限り、通常のメモリアクセス命令と同等の速度でロック操作が行なえる特長を持つ。

【 0 0 4 9 】
【発明の効果】 本発明の第1の効果は、オペレーティングシステムを経ないユーザレベルの割り込み機構を実現できることである。これによりスレッド切り替えのためのオーバーヘッドが小さくなり、従来より小さな粒度でのマルチスレッド実行が可能になる。

【 0 0 5 0 】 本発明の第2の効果は、比較的応答速度の遅い主記憶メモリをアクセスすること無くプロセッサエレメント間の併用制御が実現できることである。これに

よりスレッド管理やスレッド間の通信にかかるオーバーヘッドが小さくなり、従来より小さな粒度でのマルチスレッド実行が可能になる。

【 0 0 5 1 】 これらの低オーバーヘッド効果により、共有メモリ型マルチプロセッサ計算機の応用分野が拡大され、大きな粒度での並列性が少ないプログラムに対してマルチスレッド実行による性能向上が可能になる。

【 図面の簡単な説明】
【 図1 】 本発明が実施される計算機の構成例を示す図である。

【 図2 】 本発明の第1の実施の形態を示す図である。

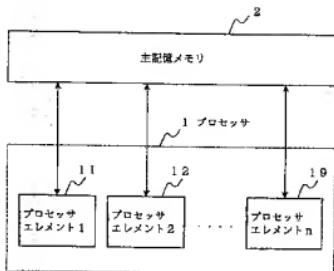
【 図3 】 本発明における高速ロック機構の第2の実施の形態を示す図である。

【 図4 】 本発明における高速ロック機構の第3の実施の形態を示す図である。

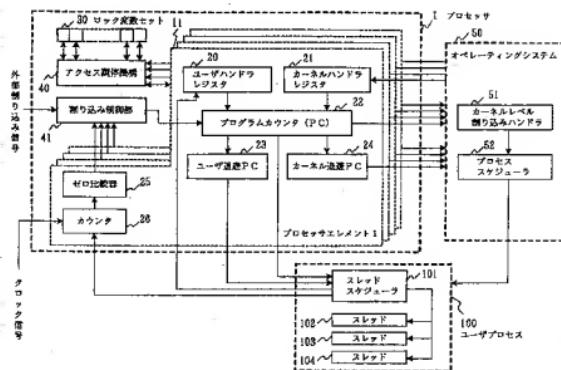
【 符号の説明】

- 1 プロセッサ
- 2 主記憶メモリ
- 1 1 ~ 1 9 プロセッサエレメント
- 2 0 ユーザハンドラレジスタ
- 2 1 カーネルハンドラレジスタ
- 2 2 プログラムカウンタ
- 2 3 ユーザ遮断PC
- 2 4 カーネル遮断PC
- 2 5 ゼロ比較器
- 2 6 カウンタ
- 3 0 ロック変数セット
- 4 0 アクセス調停機構
- 4 1 割り込み制御部
- 5 0 オペレーティングシステム
- 5 1 カーネルレベル割り込みハンドラ
- 5 2 プロセススケジューラ
- 6 0 キュー制御機構
- 6 1 ~ 6 9 キュー
- 7 0 アクセス調停機構
- 7 1 キャッシュメモリ
- 1 0 0 ユーザプロセス
- 1 0 1 スレッドスケジューラ
- 1 0 2 ~ 1 0 4 スレッド

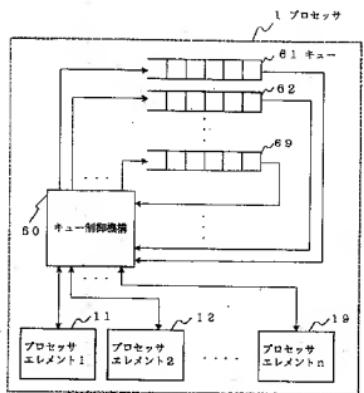
【 図1 】



【 図2 】



【 図3 】



【 図4 】

